

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

XÂY DỰNG CÁC HỆ THỐNG NHÚNG

ĐỀ TÀI: Beacon trong điều hướng các điểm du lịch

Giảng viên: Đỗ Tiến Dũng

Nhóm môn học: 02

Nhóm BTL: 09

Thành viên:

Nguyễn Công Nhật Minh

Lê Duy Mạnh

Nguyễn Vĩnh Tú

B20DCCN435

B20DCCN423

B19DCCN601

Hà Nội - 2024

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn tới thầy hướng dẫn Đỗ Tiến Dũng, thầy đã tận tình hướng dẫn giúp đỡ, chỉnh sửa và chỉ bảo nhóm chúng em trong suốt quá trình nghiên cứu và hoàn thành đề tài này.

Mặc dù đã cố gắng hoàn thành bài tập lớn này nhưng chắc chắn sẽ không tránh khỏi những sai sót, chúng em kính mong nhận được sự thông cảm và góp ý của thầy và các bạn để bài tập lớn này được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

Mục lục

I. Giới thiệu chung.....	4
II. Lý do chọn đề tài.....	5
III. Các thiết bị phần cứng.....	5
IV. Luồng hoạt động của hệ thống.....	8
V. Demo Layout ứng dụng.....	8
VI. Demo Code.....	14
VII. Công việc tuần.....	17

I. Giới thiệu chung:

- Hệ thống nhúng là hệ thống máy tính mini có khả năng hoạt động độc lập mà không cần tới máy tính để bàn hoặc mạng máy tính. Nó thường được tích hợp trong các thiết bị vật lý như màn hình điện tử, máy ATM, máy đo nhiệt độ trong bệnh viện...
- Trong du lịch, hệ thống nhúng đang được ứng dụng rộng rãi như: Máy bán vé tự động, màn hình thông tin du lịch tại các điểm đến, máy tự phục vụ thức ăn/đồ uống, máy hỗ trợ phương tiện giao thông công cộng...
- Các hệ thống nhúng trong du lịch thường được trang bị các tính năng như: Hiển thị thông tin du lịch, bán vé online, thanh toán không dùng tiền mặt, định vị và hướng dẫn du khách... phục vụ tốt hơn trải nghiệm du lịch.
- Việt Nam đang nỗ lực ứng dụng hệ thống nhúng trong du lịch nhằm nâng cao chất lượng dịch vụ, tối ưu hóa chi phí và tự động hóa quy trình. Nhiều điểm đến như Hà Nội, TP.HCM, Đà Nẵng... đã lắp đặt mạng lưới nhúng phục vụ du khách.
- Triển vọng phát triển hệ thống nhúng trong du lịch Việt Nam ngày càng rộng khắp khi các công nghệ mới được ứng dụng. Điều này sẽ nâng cao hiệu quả hoạt động du lịch.
Ble Beacon Navigation là một hệ thống định vị sử dụng công nghệ Bluetooth Low Energy (BLE) có thể ứng dụng trong việc điều hướng cho du khách khi di chuyển giữa các điểm du lịch. Một số thông tin chính về hệ thống này:
- Hệ thống sử dụng các thiết bị BLE Beacon nhỏ gọn được lắp đặt tại các điểm du lịch, vị trí quan trọng. Chúng phát tín hiệu Bluetooth để xác định vị trí.
- Du khách sử dụng điện thoại thông minh có tích hợp ứng dụng định vị BLE. Khi điện thoại phát hiện tín hiệu từ Beacon gần đó sẽ hiển thị hướng dẫn đi lối nào để đến địa điểm đó.
- Hệ thống cho phép điều hướng trong phạm vi di chuyển giữa các điểm du lịch gần nhau, như trong khuôn viên phố đi bộ, công viên, trung tâm thương mại...
- Ưu điểm của hệ thống là chi phí thấp, triển khai nhanh chóng, người dùng dễ sử dụng, không cần kết nối Internet.

- BLE Beacon Navigation hứa hẹn giúp du khách dễ dàng tự điều hướng giữa các điểm du lịch, khám phá nhiều hơn mà không cần hướng dẫn viên.

II. Lý do chọn đề tài

- Beacon là công nghệ định vị hiện đại, phù hợp với xu hướng điện thoại thông minh và Internet vạn vật. Việc nghiên cứu và ứng dụng có ý nghĩa trong xu hướng chuyển đổi số du lịch.
- Giúp du khách di chuyển, khám phá các điểm du lịch thuận tiện, nhanh chóng hơn mà không cần nhờ đến hướng dẫn viên. Tăng trải nghiệm du lịch.
- Cải thiện hệ thống định hướng, bổ sung thông tin, kích thích khách du lịch tham quan nhiều hơn tại điểm đến.
- Hỗ trợ quản lý hoạt động đón tiếp, phục vụ du khách tốt hơn cho doanh nghiệp du lịch.
- Chi phí triển khai thấp, bền vững hơn các hệ thống định vị khác.
- Là cơ hội để ứng dụng công nghệ mới, thúc đẩy phát triển ngành du lịch theo hướng công nghệ cao.

Đây là những lý do chủ yếu khi lựa chọn đề tài này cho công tác nghiên cứu.

III. Các thiết bị phần cứng:

1. Chip beacon



Chip BLE EBYTE E73 là một module đa giao thức kích thước nhỏ, tiết kiệm năng lượng được phát triển bởi Ebyte. Dưới đây là một số thông tin về chip này:

- **Tên sản phẩm:** E73-2G4M08S1E
- **IC sử dụng:** nRF52833
- **Tần số hoạt động:** 2.360~2.500GHz
- **Giao thức hỗ trợ:** BLE 5.1, Zigbee
- **Khoảng cách truyền thông:** 120m
- **Kích thước:** 13.0*18.0mm
- **Trọng lượng:** 1.0±0.1g
- **Lõi xử lý:** ARM CORTEX-M4

Chip này sử dụng IC radio tần số nRF52833 nhập khẩu từ Nordic, hỗ trợ Bluetooth 5.1, Ble mesh, Thread, Zigbee. Nó có hiệu suất cao với lõi ARM CORTEX-M4. Đây là một lựa chọn tốt cho các ứng dụng IoT và mạng không dây

2. Hộp nhựa kỹ thuật



3. Pin cúc áo 3V



IV. Luồng hoạt động của hệ thống:

Các Beacon được gửi dữ liệu phát qua BLE , dữ liệu được đọc bởi thiết bị người dùng thông qua Web/App để hiển thị các thông tin về địa điểm, đồng thời app định hướng các đường thông qua vị trí định vị của các Beacon

V. Demo layout ứng dụng

1. Giao diện đăng nhập



← Trang đăng nhập

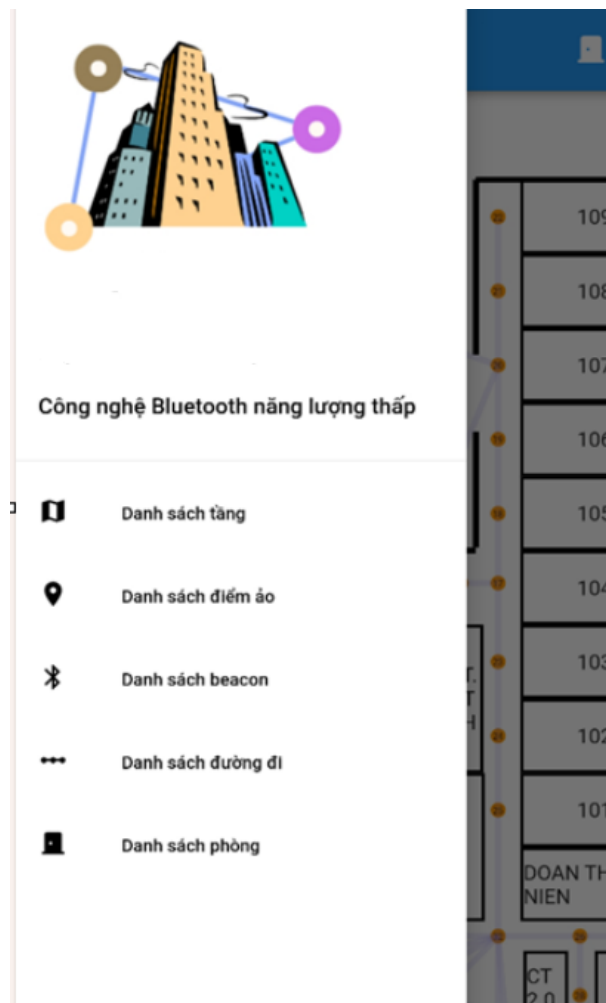

IPS CIT

Tên đăng nhập
admin

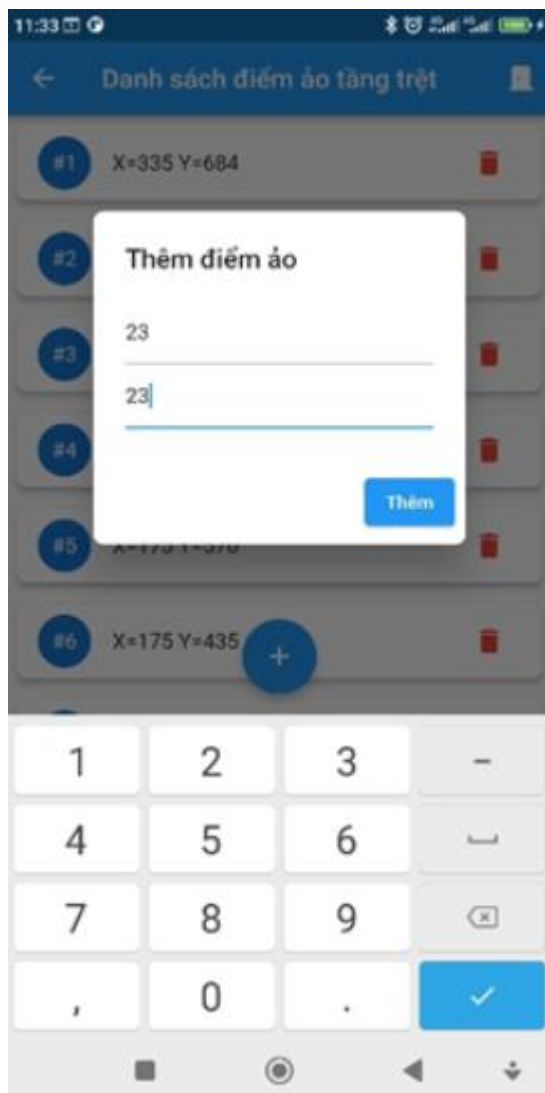
Mật khẩu

Đăng nhập

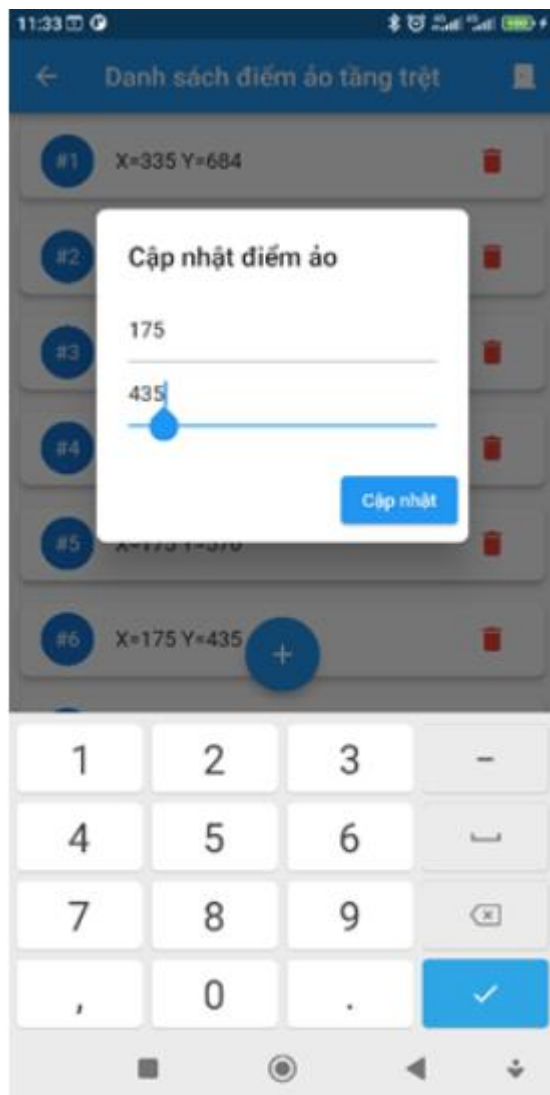
2. Giao diện tab



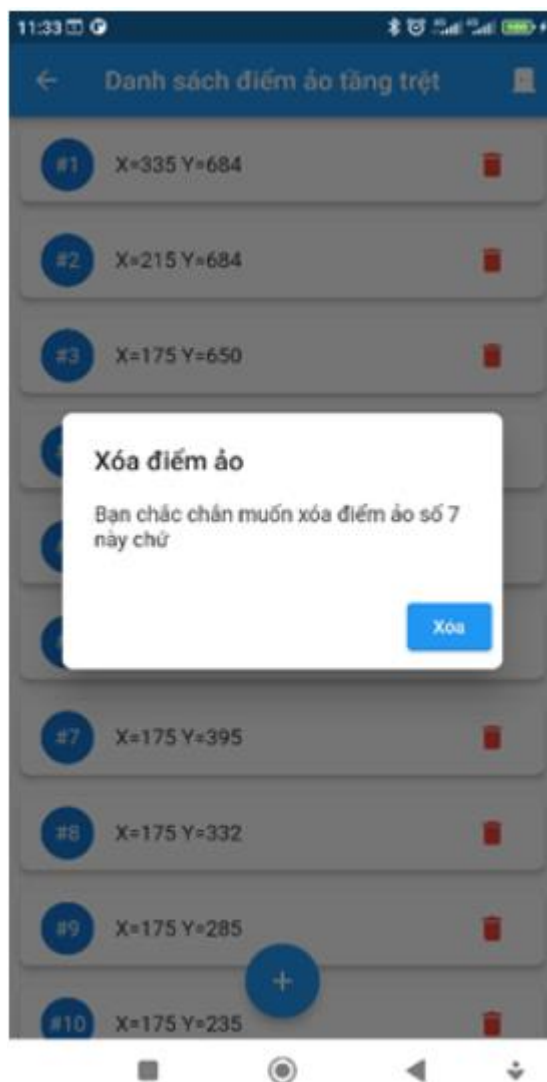
3. Giao diện thêm điểm ảo



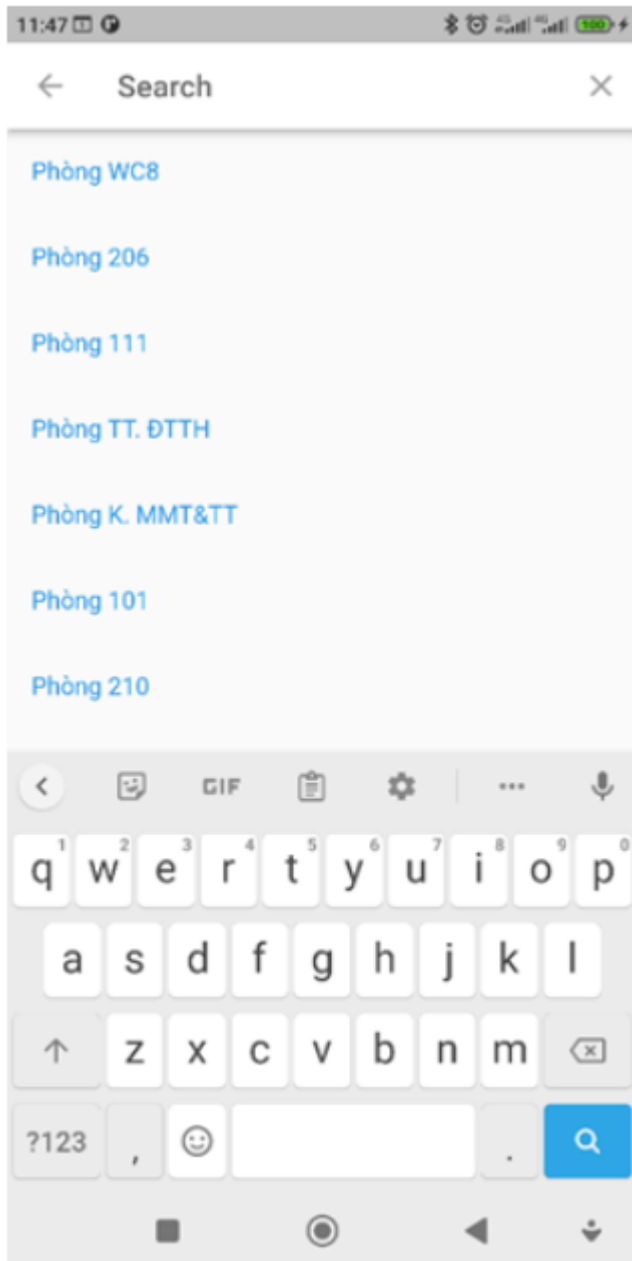
4. Giao diện cập nhật điểm ảo



5. Giao diện xóa điểm ảo



6. Giao diện Search



VI. Demo Code

```

@override
void initState() {
  debugPrint(widget.jsonFloorplan);
  load(widget.jsonFloorplan);
  super.initState();

  bleController = Get.put(BLEResult());
  //animation duration 1 seconds
  controller = AnimationController(
    duration: const Duration(milliseconds: 500), vsync: this)
    ..addListener(() => setState(() {}))
    ..forward()
    ..addStatusListener((status) {
      if (status == AnimationStatus.completed) {
        controller.reverse();
      } else if (status == AnimationStatus.dismissed) {
        controller.forward();
        centerXList = bleController.selectedCenterXList;
        centerYList = bleController.selectedCenterYList;
        // initialize radius list tao r
        radiusList = [];
        for (int i = 0; i < bleController.selectedDistanceList.length; i++) {
          radiusList.add(0.0);
        }
        // rssi to distance
        for (int idx = 0;
          idx < bleController.selectedDistanceList.length;
          idx++) {
          var rssi = bleController
            .scanResultList[bleController.selectedDeviceIdxList[idx]].rssi;
          var alpha = bleController.selectedRSSI_1mList[idx];
          var constantN = bleController.selectedConstNList[idx];
          var distance = logDistancePathLoss(
            rssi.toDouble(), alpha.toDouble(), constantN.toDouble());
          radiusList[idx] = distance;
        }
      }
    });
}

```

Code điều khiển Beacon

```

class BLEResult extends GetxController {
    // Raw BLE Scan Result
    List<ScanResult> scanResultList = [];

    // BLE advertising packet format
    List<String> deviceNameList = [];
    List<String> macAddressList = [];
    List<String> rssiList = [];

    List<String> txPowerLevelList = [];
    List<String> manufacturerDataList = [];
    List<String> serviceUuidsList = [];

    // BTN flag
    List<bool> flagList = [];

    // selected beacon param for distance
    List<int> selectedDeviceIdxList = [];
    List<String> selectedDeviceNameList = [];
    List<num> selectedConstNList = [];
    List<int> selectedRSSI_1mList = [];
    List<double> selectedCenterXList = [];
    List<double> selectedCenterYList = [];
    List<num> selectedDistanceList = [];

    // max distance
    double maxDistance = 8.0;

    // distance value
    List<double> distanceList = [];
    List<BeaconElement> beaconsDB = [];
    void parseBeaconFromResult(List<ScanResult> scanResultList,
        Future<List<BeaconElement>> beacons) async {
        beaconsDB = await beacons;
        for (ScanResult r in scanResultList) {
            for (int i = 0; i < beaconsDB.length; i++) {
                if (r.device.id.id == beaconsDB[i].macAddress) {
                    print(r.rssi.toString());
                    rssiList.add(r.rssi.toString());
                    print(beaconsDB[i].x.toString() + ":" + beaconsDB[i].y.toString());
                    selectedCenterXList.add(beaconsDB[i].x);
                    selectedCenterYList.add(beaconsDB[i].y);
                    selectedDistanceList.add(0.0);
                }
            }
        }
    }
}

```

Code định vị beacon

```

Widget buildBeaconElement(BuildContext context, BeaconElement element) {
  return Positioned(
    top: element.y - element.radius,
    left: element.x - element.radius,
    child: Container(
      height: element.radius * 2,
      width: element.radius * 2,
      decoration: const BoxDecoration(
        color: Color.fromARGB(255, 72, 171, 228),
        shape: BoxShape.circle,
      ), // BoxDecoration
    ), // Container
  ); // Positioned
}

Widget buildElement(BuildContext context, BaseElement element) {
  switch (element.runtimeType) {
    case DeskElement:
      return buildDeskElement(context, element as DeskElement);

    case RectElement:
      return buildRectElement(context, element as RectElement);

    case BeaconElement:
      return buildBeaconElement(context, element as BeaconElement);

    default:
      throw Exception('Invalid element type: ${element.runtimeType}');
  }
}

```

Code định vị Beacon trên map

```

void parseBeaconFromResult(List<ScanResult> scanResultList,
  Future<List<BeaconElement>> beacons) async {
  beaconsDB = await beacons;
  for (ScanResult r in scanResultList) {
    for (int i = 0; i < beaconsDB.length; i++) {
      if (r.device.id.id == beaconsDB[i].macAddress) {
        print(r.rssi.toString());
        rssiList.add(r.rssi.toString());
        print(beaconsDB[i].x.toString() + ":" + beaconsDB[i].y.toString());
        selectedCenterXList.add(beaconsDB[i].x);
        selectedCenterYList.add(beaconsDB[i].y);
        selectedDistanceList.add(0.0);
      }
    }
  }
}

```

Code set up khoảng cách giữa các Beacon từ dữ liệu parse

VII. Công việc tuần

STT	Công việc	Thành viên
1	Set up phần cứng, Điều chỉnh thông số cho beacon để thiết bị đọc được	Nguyễn Công Nhật Minh
2	Design luồng hoạt động, khởi tạo dự án, kết nối demo từ bluetooth của beacon gửi đến	Lê Duy Mạnh
3	Code layout của ứng dụng	Nguyễn Vĩnh Tú